In the Claims:

Please amend Claims 1, 2, 6, 11, 12 and 16; cancel Claims 7, 9-10, 17 and 19-20; and add new Claims 21-26, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented claims in a continuing or future application.

1. (Currently Amended) A system including a web-based interface for use with ~~a JMS mark-up language~~ an application program interface, comprising:

a computer including a processing device and a client operating thereon;

a web application including a user interface that executes on ~~a client machine~~ the client and allows a user to enter markup language ~~components~~ commands and communicate said markup language components to a remote server for processing thereon;

a command processor that executes on a remote server, that receives and validates the markup language commands, and, for each markup language command converts the markup language command into a command object for communication to a command dispatcher;

a command dispatcher that executes on the remote server and that receives command objects from the command processor and, for each command object, assigns the command object to one of a plurality of categories corresponding to a plurality of application program interfaces; and [[,]]

a plurality of processor modules that execute on the remote server, including a processor module for each category of application program interface, wherein each processor module receives the command objects assigned to its category, and performs appropriate operations against the corresponding application program interface ~~a command processor that executes at a remote server and converts the markup language components into one of JMS or JMX system operations~~ at the remote server.

2. (Currently Amended) The system of claim 1 wherein the markup language ~~components~~ commands are communicated as a source file, and wherein the remote server includes a parser that parses said source file to retrieve said markup language components and communicate said markup language ~~components~~ commands to said command processor.

3. (Original) The system of claim 1 wherein said user interface includes a file selection device for selecting a source file to be communicated to said command processor.

4. (Original) The system of claim 1 wherein said user interface includes a Web-based form within which a user can enter markup language commands to be communicated to said command processor.

5. (Original) The system of claim 1 wherein said web application is a web browser.

6. (Currently Amended) The system of claim 1 wherein said web application communicates said markup language ~~components~~ commands to said remote server via a wide area network or the Internet.

7. (Canceled).

8. (Original) The system of claim 1 wherein the markup language is JMS markup language.

9-10. (Canceled).

11. (Currently Amended) A method of using a web-based interface with ~~a JMS mark-up language~~ an application program interface, comprising the steps of:

providing a web application including a user interface that executes on a client machine and allows a user to enter markup language ~~components~~ commands and communicate said markup language ~~components~~ commands to a remote server for processing thereon; and,

receiving said markup language ~~components~~ commands at a command processor at a remote server that validates the markup language commands, and, for each markup language command converts the markup language command into a command object;

assigning each command object to one of a plurality of categories corresponding to a plurality of application program interfaces; and

~~converting the markup language components into one of JMS or JMX system operations~~

processing the command objects using a plurality of processor modules, including a processor module for each category of application program interface, wherein each processor module receives the command objects assigned to its category, and performs appropriate operations against the corresponding application program interface at the remote server.

12. (Currently Amended) The method of claim 11 wherein the markup language ~~components~~ commands are communicated as a source file, and wherein the remote server includes a parser that parses said source file to retrieve said markup language ~~components~~ commands and communicate said markup language ~~components~~ commands to said command processor.

13. (Original) The method of claim 11 wherein said user interface includes a file selection device for selecting a source file to be communicated to said command processor.

14. (Original) The method of claim 11 wherein said user interface includes a Web-based form within which a user can enter markup language commands to be communicated to said command processor.

15. (Original) The method of claim 11 wherein said web application is a web browser.

16. (Currently Amended) The method of claim 11 wherein said web application communicates said markup language ~~components~~ commands to said remote server via a wide area network or the Internet.

17. (Canceled).

18. (Original) The method of claim 11 wherein the markup language is JMS markup language.

19-20. (Canceled).

21. (New) A computer readable medium including instructions stored thereon, which when executed cause the computer to perform the steps of:

providing a web application including a user interface that executes on a client machine and allows a user to enter markup language commands and communicate said markup language commands to a remote server for processing thereon; and,

communicating said markup language commands to a command processor at a remote server that validates the markup language commands, and, for each markup language command converts the markup language command into a command object;

assigning each command object to one of a plurality of categories corresponding to a plurality of application program interfaces; and

processing the command objects using a plurality of processor modules, including a processor module for each category of application program interface, wherein each processor module receives the command objects assigned to its category, and performs appropriate operations against the corresponding application program interface at the remote server.

22. (New) The system of claim 1 wherein at least one of the application program interfaces conforms to the Java Message Service specification.

23. (New) The system of claim 1 wherein the plurality of application program interfaces include both application program interface that conforms to the Java Message Service specification and an application program interface that conforms to the Java Management Extensions specification.

24. (New) The method of claim 11 wherein at least one of the application program interfaces conforms to the Java Message Service specification.

25. (New) The method of claim 11 wherein the plurality of application program interfaces include both application program interface that conforms to the Java Message Service specification and an application program interface that conforms to the Java Management Extensions specification.

26. (New) The computer readable medium of claim 21 wherein at least one of the application program interfaces conforms to the Java Message Service specification.